intermediate data in the memory **430**, and outputs the intermediate data to each of a plurality of write drivers **820**.

[0081] The input buffer and decoder **810** outputs bits corresponding to one pixel value to each of the write drivers **820**. If a pixel value is 8 bits, the input buffer and decoder **810** outputs each bit to eight (8) write drivers **820** and the pixel values of 8 bits are stored in eight (8) cell arrays **840**.

[0082] The write driver **820** writes the intermediate data received from the input buffer and decoder **810** to the cell arrays **840**. The write driver **820** activates cells in the column direction of the cell arrays **840** and writes the intermediate data to the active cells. Accordingly, the form of the intermediate data generated by the first core **420** and the form of the intermediate data stored in the cell arrays **840** are same. Accordingly, no separate address calculation is needed for a read sense amplifier (S/A) **830** to read out the intermediate data stored in the cell arrays **840** in the row direction. The read S/A **830** sequentially reads out the data stored in the cell arrays **840** in the row direction.

[0083] The read S/A **830** reads out the data stored in the cell arrays **840** in the row direction. The read S/A **830** may activate cells in the row direction of the cell arrays **840** and measure resistance of each cell. The read S/A **830** acquires the data stored in each cell according to the amount of a resistance value of each cell.

[0084] A MUX and output buffer **850** receives the data read out from the cell arrays **840** and outputs the received data to the second core **440**.

[0085] FIG. **9** illustrates a method of storing intermediate data in a cell array. The intermediate data is data on which 1D FFT is performed once in the row direction or in the column direction. A pixel value denotes data stored in one pixel.

[0086] When a first pixel value **901** is "10010110", the first pixel value **901** is stored in eight (8) cell arrays **910** to **980**. Each bit may be stored at the same location in the respective cell arrays. For example, the first bit "1" of the first pixel value **901** is stored in the first cell array **910**, the second bit "0" in the second cell array **920**, and the final bit "0" in the eighth cell array **980**.

[0087] When a second pixel value **902** is "00111011", the second pixel value **902** is stored in the eight cell arrays **910** to **980**. When the second pixel value **902** is a pixel value located at the right of the first pixel value **901**, each bit of the second pixel value **902** is stored in a cell at the right of each bit of the first pixel value **901**. As illustrated in FIG. **9**, the first bit "0" of the second pixel value **902** is stored in the first cell array **910** at the right of the first bit of the first pixel value **901**, the second bit "0" in the second cell array **920** at the right of the second bit of the first pixel value **901**, and the final bit "1" in the eighth cell array **980** at the right of the final bit of the first pixel value **901**.

[0088] While not restricted thereto, an exemplary embodiment can be embodied as computer-readable codes in a computer-readable recording medium that includes program instructions to be implemented by a computer to cause a processor to execute or perform the program instructions. The medium may also include, alone or in combination with the program instructions, data files, data structures, and the like. The computer-readable recording medium may include any kind of recording devices for storing data which can be thereafter read by a computer system. Examples of the computer-readable recording medium include magnetic storage media (e.g., ROM, RAM, floppy disks, hard disks, etc.), and optical recording media (e.g., CD-ROMs, digital versatile disks (DVDs), etc.). The described hardware devices may be configured to act as one or more software modules in order to perform the operations and methods described above, or vice versa. In addition, the computer-readable recording medium may be distributed over network coupled computer systems so that the computer-readable code is stored and executed in a distributive manner.

[0089] Also, an exemplary embodiment may be written as a computer program transmitted over a computer-readable transmission medium, such as a carrier wave, and received and implemented in general-use or special-purpose digital computers that execute the programs. Moreover, it is understood that in exemplary embodiments, one or more units of the above-described apparatuses and devices can include circuitry, a processor, a microprocessor, etc., and may execute a computer program stored in a computer-readable medium.

[0090] At least one of the components, elements or units represented by a block as illustrated in the above diagrams may be embodied as various numbers of hardware, software and/or firmware structures that execute respective functions described above, according to an exemplary embodiment. Also, at least one of these components, elements or units may be specifically embodied by a module, a program, or a part of code, which contains one or more executable instructions for performing specified logic functions. Also, at least one of these components, elements or units may further include a processor such as a central processing unit (CPU) that performs the respective functions, a microprocessor, or the like. The exemplary embodiments may be described in terms of functional block components and various processing steps. Such functional blocks may be realized by any number of hardware and/or software components configured to perform the specified functions. For example, the exemplary embodiments may employ various integrated circuit (IC) components, e.g., memory elements, processing elements, logic elements, look-up tables, and the like, which may carry out a variety of functions under the control of one or more microprocessors or other control devices. Similarly, where the elements of the exemplary embodiments are implemented using software programming or software elements, the exemplary embodiments may be implemented with any programming or scripting language such as C, C++, Java, assembler language, or the like, with the various algorithms being implemented with any combination of data structures, objects, processes, routines or other programming elements. Functional aspects of the above exemplary embodiments may be implemented in algorithms that are executed on one or more processors. Furthermore, the exemplary embodiments may employ any number of related art techniques for electronics configuration, signal processing and/or control, data processing, and the like. The words "mechanism," "element," "means," and "configuration" are used broadly and are not limited to mechanical or physical embodiments, but may include software routines in conjunction with processors, etc.

[0091] The exemplary embodiments shown and described herein are illustrative examples of the inventive concept and are not intended to otherwise limit the scope of the inventive concept. For the sake of brevity, related art electronics, control systems, software development and other functional aspects of the systems may not be described in detail. Furthermore, the connecting lines, or connectors shown in